CIET CHATBOT

A PROJECT REPORT

Submitted by

BALAJI B	**71052002005**
HARISH KUMAR P	**71052002017**
NAGARAJ M	**71052002032**
SARAN S	**71052002048**

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

COIMBATORE INSTITUTE OF ENGINEERING AND TECHNOLOGY

COIMBATORE - 641 109

MAY 2023

BONAFIDE CERTIFICATE

Certified that this project report “ **CIET CHATBOT** ” is the bonafide work of **“BALAJI B,HARISHKUMAR P,NAGARAJ M,SARAN S”** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. K. PUSHPALATHA M.E., Ph. D.,	**Dr. K. PUSHPALATHA M.E., Ph. D.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Associate Professor	Associate Professor
Department of Computer Science	Department of Computer Science
and Engineering	and Engineering
Coimbatore Institute of	Coimbatore Institute of
Engineering and Technology	Engineering and Technology
Coimbatore – 641109	Coimbatore – 641109

Submitted for the university examination held on ..………….. at Coimbatore Institute of Engineering and Technology, Coimbatore - 641109

INTERNAL EXAMINER	EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We, have great and immense pleasure in expressing the acknowledgment to the numerous contributors for the success of this project work. They all deserve credit and thanks to carry out this project successfully.

We would like to record our sincere indebtedness and gratitude to our beloved Director**, Dr.K.A.Chinnaraju M.Sc.,M.B.A.,Ph.D.,** for his noteworthy efforts to enhance our professional dexterity and co-curricular excellence.

We are grateful to our respected Principal**, Dr.N.Nagarajan, M.E., Ph.D.,** for providing us with necessary facilities to carry out our project work.

We are very much pleased to acknowledge our sincere thanks to our Head of the Department and guide **Dr.K.Pushpalatha M.E.,Ph.D.,** for her guidance and valuable suggestions.

Finally, We extend our thanks to the management, faculty members, parents and our student friends for their support and encouragement and to all others, who extended their helping hands to us in the completion of our final year project.

TABLE OF CONTENTS

ABSTRACT

The chatbot is a natural language processing (NLP) project that aims to simulate human conversation and provide an automated response to user input. The project involves training a chatbot model using a dataset of user queries and responses, and implementing the model in a web application to enable users to interact with the chatbot. The chatbot is designed to understand user queries, identify the intent behind the query, and generate an appropriate response. The project also involves pre-processing user input, tokenizing the input, and removing stop words to enhance the accuracy of the model. Additionally, the chatbot incorporates machine learning algorithms and artificial intelligence techniques to improve its responses over time. The chatbot project has many practical applications, including customer service, technical support, and online shopping assistance.

AI is considered in chatbot development because it allows the chatbot to simulate a human-like conversation by understanding natural language input and generating appropriate responses. AI techniques like natural language processing (NLP) and machine learning (ML) enable chatbots to learn from user interactions and improve over time. Additionally, AI enables chatbots to perform tasks beyond basic question-answering, such as recommending products, scheduling appointments, and providing personalized assistance. Overall, AI plays a crucial role in making chatbots intelligent and capable of providing effective and engaging user experiences.

LIST OF TABLE

LIST OF FIGURES

LIST OF ABBERVATIONS

AI	-	Artificial Intelligence
NLP	-	Natural Language Processing
NLU	-	Natural Language Understanding
GAN	-	Generative Adversarial Networks
CORS	-	Cross Origin Resource Sharing
API	-	Application Programming Interface
NLTK	-	Natural Language Tool Kit
SIS	-	Student Information System
LMS	-	Learning Management System
FAQs	-	Frequently Asked Questions
UI	-	User Interface
RL	-	Reinforcement Learning
ML	-	Machine Learning

CHAPTER 1

INTRODUCTION

1.1 Overview

1.1.1 Artificial Intelligence :

Artificial Intelligence (AI) is a field of computer science and engineering that aims to create intelligent machines that can perform tasks that typically require human intelligence, such as recognizing speech, understanding natural language, making decisions, and learning from experience. AI involves the development of algorithms and computer programs that can process information, reason, and learn, as well as the use of large amounts of data to train and improve machine performance.

AI can be classified into several subfields, including machine learning, deep learning, natural language processing, computer vision, robotics, and expert systems. Machine learning, which involves training machines to learn patterns and make predictions based on data, is a particularly important subfield of AI. Deep learning, which is a subset of machine learning that uses artificial neural networks to learn and improve performance, is also becoming increasingly popular.

AI has many practical applications, including in healthcare, finance, transportation, education, and entertainment. It is also being used to develop autonomous vehicles, smart homes, and other advanced technologies that can improve quality of life and increase productivity. However, AI also raises ethical and social issues related to privacy, bias, and job displacement, which must be carefully considered and addressed.

One of the key goals of AI is to create intelligent systems that can learn from experience and improve their performance over time. This is typically done through the use of machine learning algorithms, which allow machines to learn from data and make predictions or decisions based on that learning. There are many different types of machine learning algorithms, including supervised learning, unsupervised learning, and reinforcement learning.

Other important areas of AI research include natural language processing (NLP), which focuses on enabling machines to understand and generate human language, computer vision, which involves giving machines the ability to perceive and understand visual information, and robotics, which involves creating machines that can interact with and manipulate the physical world.

AI has a wide range of practical applications, including in fields such as healthcare, finance, transportation, and manufacturing. Some examples of AI applications include personalized medicine, fraud detection, autonomous vehicles, and predictive maintenance.

Overall, AI is a rapidly evolving field that is transforming many aspects of our lives and has the potential to revolutionize many industries in the years to come.

1.1.2 Approaches in Artificial Intelligence :

There are several approaches in artificial intelligence, which are as follows:

1. Rule-based approach: This approach involves defining a set of rules for the AI system to follow. The rules are generally based on the knowledge and expertise of human experts in a particular field. For example, an AI system that helps diagnose diseases may use a rule-based approach to match a patient's symptoms to a known set of symptoms associated with different diseases.

2. Machine learning approach: This approach involves training an AI system to recognize patterns and make decisions based on that pattern recognition. Machine learning algorithms can be trained using large sets of data, and they can learn to identify complex patterns that might be difficult or impossible for humans to recognize.

3. Deep learning approach: Deep learning is a subset of machine learning that uses neural networks to learn from large sets of data. Neural networks are a set of algorithms that are modeled after the structure and function of the human brain. Deep learning algorithms can learn to recognize and classify images, speech, and other types of data with very high accuracy.

4. Evolutionary approach: This approach involves using genetic algorithms to evolve solutions to complex problems. In this approach, the AI system starts with a population of candidate solutions to a problem, and it uses a process of selection, mutation, and crossover to generate new, improved solutions over time.

5. Hybrid approach: Many AI systems use a combination of these approaches, combining rule-based systems with machine learning or deep learning algorithms to achieve the best results. Hybrid approaches can be particularly effective for complex tasks, where different approaches are needed handle different aspects of the problem.different aspects of the problem. .

1.1.3 Learning in AI :

Learning is a crucial aspect of AI, and it refers to the ability of a system to acquire knowledge or skills through experience or training. There are several different types of learning that are used in AI:

1. **Supervised learning:** In supervised learning, a model is trained on a labeled dataset, where each example is accompanied by a desired output. The model learns to predict the output for new inputs by adjusting its parameters to minimize the difference between its predictions and the desired outputs.

2. **Unsupervised learning**: In unsupervised learning, a model is trained on an unlabeled dataset, where the desired output is not known. The model learns to discover patterns or structure in the data by clustering or dimensionality reduction techniques.

3. **Reinforcement learning:** In reinforcement learning, an agent interacts with an environment and learns to take actions that maximize a reward signal. The agent learns by trial and error, adjusting its actions based on the feedback it receives from the environment.

4. **Transfer learning:** In transfer learning, a model is trained on one task and then applied to another task. The knowledge learned in the first task is transferred to the second task, which can help the model learn more quickly and with less data.

5. **Semi-supervised learning:** In semi-supervised learning, a model is trained on a combination of labeled and unlabeled data. The model learns from the labeled data to make predictions on the unlabeled data, which can improve its performance on both labeled and unlabeled examples.

These learning approaches can be used alone or in combination with each other, depending on the specific problem being solved and the available data. The goal of learning in AI is to develop models that can generalize well to new, unseen data and make accurate predictions or decisions.

1.1.3 Chatbot :

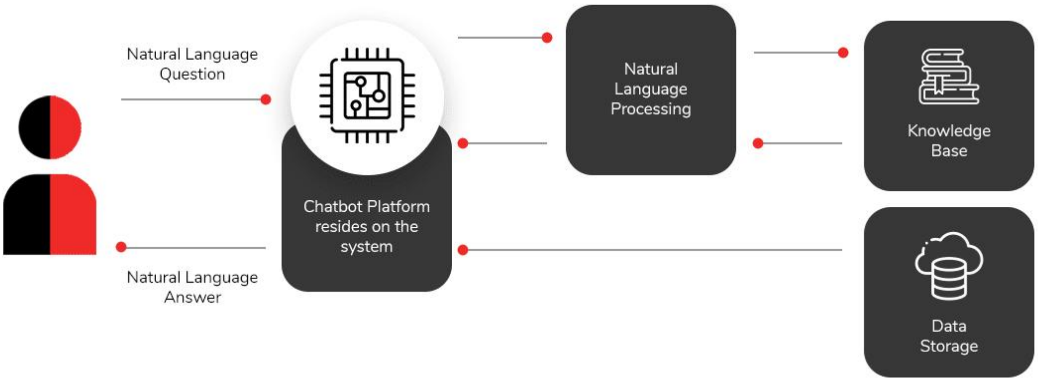
A chatbot is an AI-powered computer program that simulates conversation with human users, typically through text or voice interactions. Chatbots can be programmed to perform a variety of tasks, such as answering frequently asked questions, providing customer support, assisting with online purchases, or simply engaging in casual conversation.

Chatbots are often used in customer service settings, where they can handle simple and repetitive inquiries and free up human agents to focus on more complex issues. They can also be used in marketing and sales, where they can provide personalized recommendations and product information to potential customers.

In addition to their commercial applications, chatbots can also be used in healthcare, education, and other domains. For example, a chatbot might be programmed to provide medical advice or therapy to patients, or to assist with language learning exercises.

Chatbots can be designed in a variety of ways, ranging from simple scripted bots that follow a predetermined set of responses to more complex AI-powered bots that use natural language processing and machine learning to understand and respond to user queries. The effectiveness of a chatbot depends on its design, the quality of its programming, and the specific use case for which it is intended.

1.1 Working process of chatbot


Natural Language Question
Natural Language Answer
Chatbot Platform resides on the system
Natural Language Processing
Knowledge Base
Data Storage

1.2. Ciet Chatbot:

1.2.1 Problem Statement :

Our college lacks an efficient and interactive way of addressing the queries and concerns of students, faculty, and staff members. Due to this, there is often a lack of clarity and transparency in communication, leading to confusion and frustration. Therefore, there is a need to develop and implement a chatbot that can provide prompt and accurate responses to common queries and concerns and assist in providing a better overall experience for the college community.

1.2.2 Objective :

The objective of a chatbot is to provide automated assistance to users in a conversational manner. Chatbots are typically designed to simulate human conversation using natural language processing (NLP) and machine learning algorithms.

The main goals of a chatbot include:

1. Improving customer engagement: Chatbots can help businesses engage with their customers by providing personalized and timely responses to their inquiries.

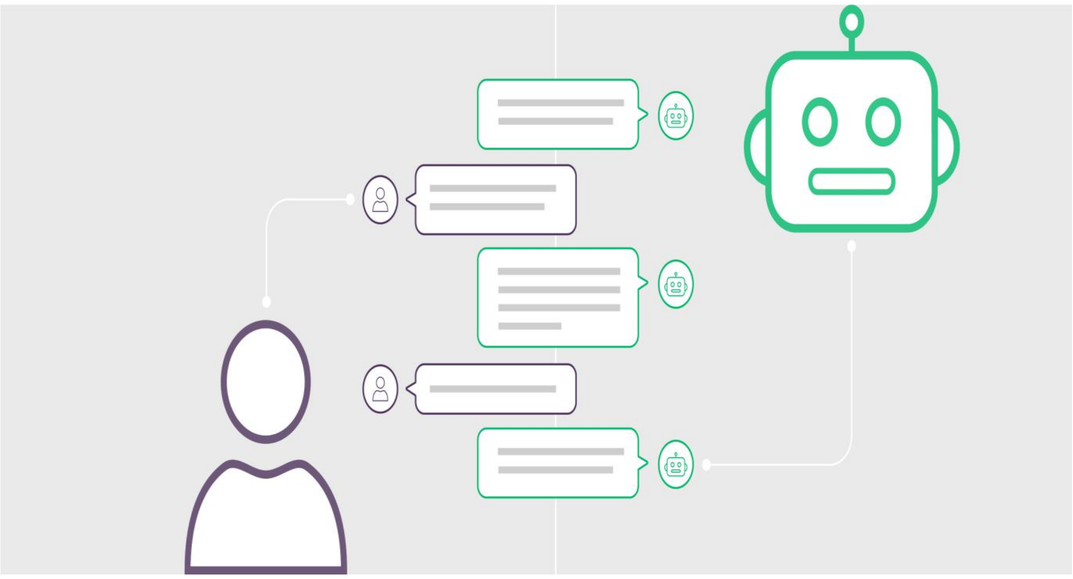
2. Enhancing customer experience: Chatbots can provide customers with 24/7 assistance, which can improve their overall experience with a brand.

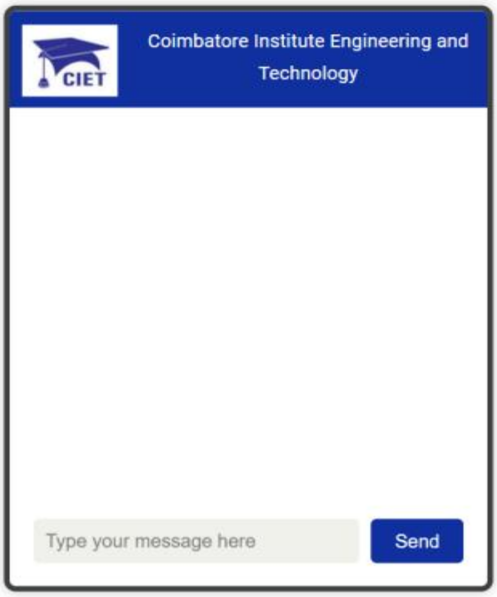
3. Increasing efficiency: Chatbots can automate repetitive tasks, allowing businesses to save time and reduce costs.

4. Generating leads and sales: Chatbots can be programmed to promote products or services, and even take orders from customers.

Overall, the objective of a chatbot is to improve communication and interaction between businesses and their customers, while also increasing efficiency and reducing costs.

1.2 Chatbot User Interface.





1.2.3 Algorithms :

Among other things, some of the most popular algorithms used by conventional Chatbots are Naïve Bayes, Decision Trees, Support Vector Machines, Markov Chains, and Natural Language Processing (NLP).

Table 1.1 Algorithms

Algorithms	**Description**
Navie Bayes	Naïve Bayes algorithm attempts to classify text into certain categories so that the chatbot can identify the intent of the user, and thereby narrowing down the possible range of responses. Since intent identification is one of the first and foremost steps in chatbot conversations, it is imperative that this algorithm works properly.
Support Vector Machines	SVMs work based on the principle of Structural Risk Minimization Principle. SVMs work very well with text data and Chatbots because of the high dimensional input space due to large number of text features, linearly separable data and the prominence of sparse matrix. It is one of the most popularly used algorithms for text classification and intent identification.
Deep Neural Networks	Inspired by the human brain, neural networks consist of layers of interconnected artificial neurons which communicate with one another. These neurons learn features from the data and work together to produce a meaningful output. Neural networks are data intensive and require huge volumes of data to learn patterns and trends in the data. To test this algorithm, we should determine whether the chatbot generates valid response to

Algorithms	Description
	inputs, keeps the conversation flowing, addresses the need of the user and whether the chatbot is able to mimic the linguistic characteristics of a human to a reasonable extent. This may mean an adaptation of the Turing test may be an appropriate testing method
Markov Chains	The Markov Chain algorithm is based on the concept of a Markov process, which is a stochastic process in which the probability of a future state depends only on the current state and not on any previous states.
Natural Language Processing(NLP)	NLP stands for Natural Language Processing, which is a subfield of artificial intelligence (AI) that focuses on enabling machines to understand, interpret, and generate human language. NLP involves several techniques such as text mining, machine learning, and computational linguistics, which are used to process and analyze large amounts of natural language data such as text, speech, and images.

Natural Language Processing (NLP):

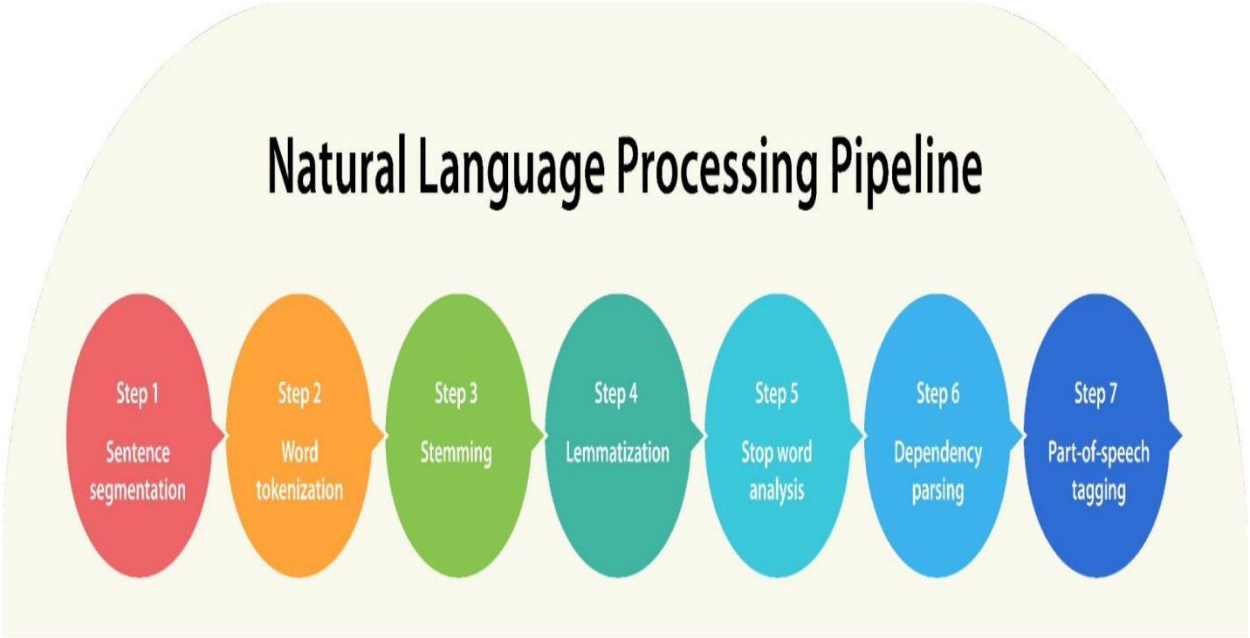
NLP is considered one of the best technologies for developing a chatbot when it is compared with others because it enables the chatbot to understand natural language inputs and generate human-like responses. NLP allows the chatbot to interpret the user's intent and respond in a personalized and contextual manner. The application of computational techniques to the analysis and synthesis of natural language and speech.

With NLP, chatbots can understand the nuances of human language, including slang, idioms, and colloquialisms, which makes them more effective in communicating with users. Chatbots powered by NLP can also learn from past interactions and improve their responses over time, making them more intelligent and capable of handling complex tasks.

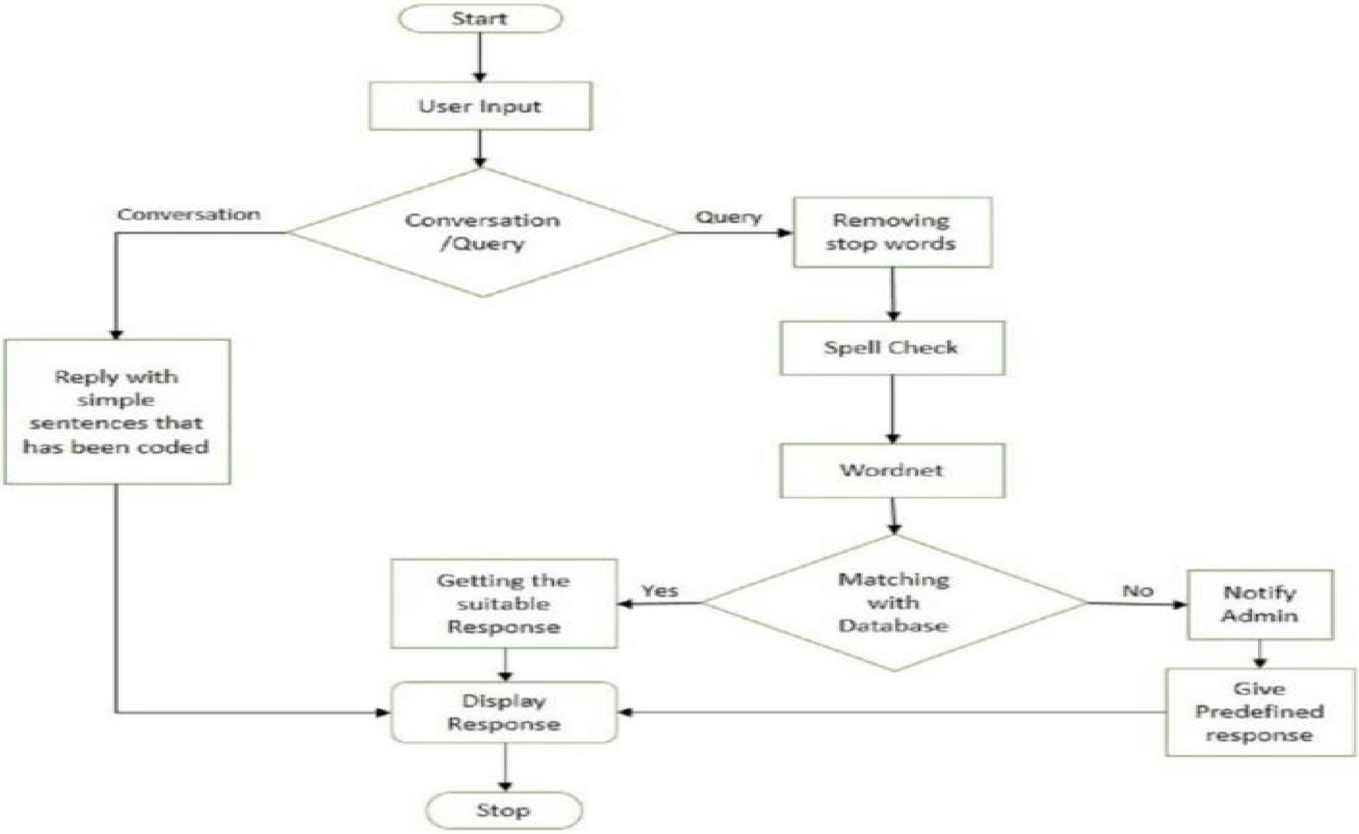
Furthermore, NLP can be used to integrate chatbots with other technologies such as voice assistants, messaging apps, and social media platforms. This allows businesses to provide seamless and personalized customer support across multiple channels.

Overall, NLP is the best technology for developing a chatbot because it enables the chatbot to understand natural language inputs, generate human-like responses, and learn from past interactions, making it more effective and intelligent in handling complex tasks.

1.3 The Natural Language Processing Pipeline



1.4 Chatbot workflow



CHATPER 2
LITERATURE SURVEY

2.1 A Review on College Enquiry Chatbot

The College Enquiry Chat Bot project is an Android application that provides students with information about college activities and updates through an effective graphical user interface. The bot uses artificial intelligence algorithms to understand and respond to user queries about college events and important notices. The system also includes an online notice board where important notices can be displayed, making it easier for students to stay up-to-date. The bot matches keywords in the knowledge base to answer user queries and if a relevant answer is not found, the user can mark it as invalid, which is sent to the Admin panel. The aim of this project is to provide an efficient and convenient way for students to make enquiries without having to physically go to the college.

2.2 Chabot for university-related FAQs , 2017 International Conference on Advances in Computing, Communications, and Informatics.

Today's era is having many web based services like E-business, Entertainment, Virtual assistance and many more. There is drastic increase in the world of web service, where every thing is now getting associated with web. It is very user friendly approach to avail everything to doorstep. There are different types of customer service available like live chat support service, phone(telephone) services. But for all such support services provided by human to human takes time to answer customers query. As the number of clients increases the waiting time increases as well, which results in poor client satisfaction. One of the important goals in the field of Human Computer Interaction (HCI) is the outline of normal and instinctive connection modalities. Specifically, numerous endeavors have been committed to the improvement of frameworks to communicate with the client in a characteristic language . Computer based chatbots are getting to be distinctly famous as an intuitive and successful open framework between human and machines. Chatbot is a manufactured substance that is intended to reproduce a clever discussion with human accomplices through their regular language. Currently, chatbots are utilized by a great many web clients to intercede access to information or learning bases and furthermore to do non specific discussions . AIML and LSA is used for creating chatbots. AIML is Artificial Intelligence Markup Language (AIML) and Latent Semantic Analysis (LSA) are used for developing chatbots, which are used to define general queries like how do you do?, how can I help you etc. This pattern can also be used to give random responses for same query. LSA is a Latent Semantic Analysis,which is utilized to discover likenesses between words as vector representation. So that the unanswered queries by AIML will be viewed as a reply by LSA. Most chatbots basically search for keywords, phrases, and examples that have been customized into their databases, yet some utilize more propelled strategies. So far no chatbot has possessed the capacity to totally trick people into trusting it as one of them through its information of regular dialect . In this paper the need for chatbot in education domain is highlighted and designed to provide visitor satisfaction.

2.3 PCE College Enquiry Bot, Priyadarshini College of Engineering, Nagpur, India, International Journal of Innovations in Engineering and Science.

People are using messaging applications now a days, it would be great to use a smart agent which would answer all queries. CHATBOT with its multiple usage over a large number of applications such as shopping, newsupdates, ticket reservations, customer service, entertainment, enquiry systems and much more. Chatbots are fast,easily accessible and easy to use. Chat bots are virtual agents which will converse like humans are interacting. They will try to answer your questions or doubts through chatting medium as if human is interacting with you. This project deals with creation of a chatter bot which will answer College related queries quickly as possible .As a fresher or unaware person we need basic information to know about college ,this application will help in providing such enquiry information through chatting medium. Unity is basically a gaming platform which supports 2D as well as 3D gaming graphics. It is a cross platform and also used to build applications on android. Unity 5.6.3p2 is used to build this application. In making User interface unity proved to be a good application. All the designing, animation, pictures ,immersion, fontstyles, graphics buttons are built within this app. A. Dialogflow (Api.ai)– for Database management Api.ai is renamed as Dialogflow.The api.ai has SDKs which contain voice recognition, natural language understanding, and text-to-speech. Api.ai offers a web interface to build and test conversation scenarios. The platform is based on the natural language processing. Dialogflow also includes an analytics tool that can measure the engagement or session metrics like usage patterns, latency issues, etc

2.4 A Web Based College Enquiry Chatbot with Results, InternationalJournal of Innovative Research in Science, Engineering and Technology.

The article discusses the use of chatbots in the field of education, specifically in the context of helping undergraduate students with their college admissions process. The chatbot is designed to assist students in finding suitable institutions and answering their questions related to college activities. The system uses a Domain Specific Knowledge System and a Graphical User Interface to simulate a conversation with the user. The system also has an online notice board to keep users updated on important information. The chatbot responds to user questions using a Keyword Matching algorithm and allows users to mark responses as invalid, which are then reviewed by the admin. The system has two types of users, Admin and Students, with registered and unregistered student users. The use of chatbots can help relieve the burden on institutions and assist thousands of students in finding the best college options for them.

The framework being described in this paragraph is a chatbot designed to assist students in the process of choosing a college or university for their higher education. The chatbot is designed to answer questions related to college activities such as date and timing of annual day, sports day, and other social activities. It also includes an online notice board where any text notification or PDF files can be displayed to keep the user updated with important notes.

The chatbot uses a Domain Specific Knowledge System to work in the area of personal assistance required during the counseling process. It employs a "Keyword Matching" algorithm to match the user's questions with the learning base to provide relevant answers. If the match is not found, the chatbot sends an "invalid" occurrence to the Admin board, and the administrator can make necessary changes to the learning base for the next time.

The system has two types of users: Admin and Students. The Admin manages the entire system, while the Students can be either registered or unregistered. Registered students can log in using their User ID and Password to ask their questions, while unregistered students must first register themselves in the system by filling out a simple registration form before they can ask their questions.

Overall, the system aims to relieve the burden on educational institutions by providing a chatbot that can assist thousands of students in choosing the best and most appropriate institution for them.

2.5 A Web Based College Enquiry Chatbot with Results, International Journal of Innovative Research in Science, Engineering and Technology

We sometimes pass our time by chatting with different chatterboxes available on internet, which is often aimed for such purposes or just entertainment . The chatbots have embedded knowledge which helps them to identify the user’s query and give a response to it . The college enquiry chat-bot project is built using algorithms that analyses user’s queries and understand user’s message. This system is a web application which provides answer to the query of the student. Students just have to query through the bot which is used for chatting. The user can query any college related activities through the system. The user doesn’t have to personally go to the college for enquiry. The system analyses the question and then answers to the user. The system answers to the query as if it is answered by the person. With the help of algorithms, the system answers the query asked by the students. The system will also have an online notice board on which any Text notices or PDF documents can be displayed. This will help the user to be updated with the important notices. Not much time will be wasted by the user to search for the important notices.

2.6 College Enquiry Chatbot, International Research Journal of Engineering and Technology (IRJET)

A Student chatbot project is developed with the help of codeigniter that is widely called a php framework that analyzes user’s queries and perceive user’s message. The proposed System could be a internet application that provides answers to the queries provided by the scholar or the user. Users will merely question through the chatbot that's used for chatting. Students will chat by using any format there's no specific format the user needs to follow. The answers are applicable what the user queries. If the answers are found to be invalid or not accessible then those queries are hold on into the unrequited table that is essentially created by the admin. Later those queries can updated by the admin, just in case of urgency we are going to provides a message that “our representatives will get in touch with you shortly”. This may be displayed once aggregation the specified info from the user. Admin will read invalid answer through portal via login System, it'll permits the admin to get rid of the invalid answer also as in updating the acceptable answer for the question raised by the user. The User will raise any faculty connected activities through the system. The user doesn't ought to in person move to the faculty for enquiry. The System analyzes the question and so answers to the user. The system answers to the question as if it's answered by the person. The system replies with the help of a good Graphical interface which suggests that as if a true person is reprehension the user. The user will question regarding the faculty connected activities through on-line with the assistance of this internet application. this method helps the scholar to be updated regarding the college related information

2.7 Intelligent smart chatbot system for college,International Research Journal of Modernization in Engineering Technology and Science

In this paper, The aim of the Proposed System is to implement a web based bot Application, which gives responses to the query of the student very efficiently. Students have to put their queries to the bot and they are used for chatting. The system will use one the best artificial intelligence algorithms to give more efficient answers to the user. If the answer is found not appropriate, then some system to declare the answer as invalid can be incorporated. These invalid answers can be identified and they are modified by the admin of the system. Student can use the chat bot to get the answers to their queries. Our system may help students to stay updated and informed with the college activities. The incremental build model is a method of software development where the product is designed according to the user requirements, implemented and tested incrementally until the product is satisfied users reqirements. This system combines the elements of the waterfall model with the iterative philosophy of prototyping. The interaction with a Chatbot is very user friendly. It answers to the questions asked by the user. During designing a Chatbot, how does the Chatbot speak to the user? And how will be the conversation with the user and the chatbot is very important.

2.8 College Enquiry Chatbot System using Artificial Intelligence,International Journal of Scientific Research in Computer Science, Engineering and Information Technology

The proposed system is a chatbot that uses deep learning with a feed forward neural network and natural language processing. Chatbots are an advanced form of question-answering systems. The advantages of the proposed system include a simple user interface, personalized conversation, learning ability, and acknowledgement of questions beyond its scope. Deep learning is a subset of machine learning that employs algorithms to process data and imitate the thinking process, or to develop abstractions. It uses layers of algorithms to process data, understand human speech, and visually recognize objects. Feature extraction is another aspect of deep learning that automatically constructs meaningful features of data for training, learning, and understanding. Speech recognition, the conversion of voice data into text data, is required for any application that follows voice commands or answers spoken questions, but is challenging due to variations in speech.

College enquiry chat-bot is a simple web based application which aims to provide information regarding college. The chat-bot uses Natural Language Processing and Artificial Neural Network to have conversation with humans. The responses of this chat-bot are programmed up to some extent which trains the model for giving responses.

2.9 A Review on College Enquiry Chatbot,G.H. Raisoni College of Engineering and Management, Pune, India

The proposed system is a chat bot application for college students, designed to provide information about college activities and other queries related to the college. The system uses an effective graphical user interface, and the user can access various helping pages through registration and login. The chat bot uses artificial intelligence algorithms to understand and respond to user queries. The system also includes an online notice board to display important notices. The chat bot matches keywords from a knowledge base to provide relevant answers, and users can mark invalid answers for review by the admin. The system aims to provide an efficient and convenient way for students to access college-related information.

University Information Enquiry bot project will be built using artificial intelligence algorithms that will analyze user's queries and understand user's message. This system will be a web application which will provide answers to the queries of the user. Users will just have to ask the query to the bot that will be used for chatting. Artificial intelligence will be used to answer the users' queries. The user will get the appropriate answers to their queries. The answers will be giving using the built-in artificial intelligence algorithms. Students will not have to go to the college to make the enquiry. In some cases, user may find out that the answer given to his/her query is not relevant. In such cases, the user can mark this answer as Invalid, and an instance of this invalid answer will be sent to the Admin panel at the same time. Whenever Admin will log in, he will get to see the answers which are marked invalid and then he can do the necessary changes to the knowledge base so that user will get the proper result when he will ask the same query next time. The aim of our Proposed System is to develop an Android based bot Application, which provides answer to the query of the student very effectively. Students just have to put their query to the bot which is used for chatting.

2.10 Research Paper on Chatbot for Student Admission Enquiry,Journal of Advancement in Software Engineering and Testing.

A chatbot is an instance of emotional figuring system that mirrors human conversations to give instructive, esteem based, and conversational organizations. Despite that no matter how you look at it gathering, chatbots still experience the evil impacts of different execution issue as a result of imprisonments with their programming and training. Chatbots must destruction different issues, including flexibility, low-dormancy, and privacy. Today time is having numerous electronic organizations like E-business, Entertainment, Virtual assistance and some more. There is radical augmentation in the domain of web organization, where everything is right now getting related with web. It is extremely simple to utilize approach to manage benefit everything to doorstep. There are different sorts of customer help open like live talk reinforce organization, phone (telephone) organizations. In any case, for all such assistance organizations given by human to human puts aside some push to respond to customer's question. As the amount of clients develops the holding time increases as well, which realizes poor client satisfaction. One of the critical goals in the field of Human Computer Interaction (HCI) is the system of run of the mill and natural affiliation modalities. Specifically, different endeavors have been centered around the improvement of structures to talk with the client in a trademark language. The world is moving towards the automation of human effort with the help of machine learning, artificial intelligence and robotics. While there is a huge scope of improvement, some of the key areas that can be targeted for automation are responder system in education, healthcare etc. An artificial intelligence based bot can replace human efforts without compromising on the quality of response. Machine learning and artificial intelligence is used to implement chatbot along with python scripts. Past chatbots, Conversational AI suggests the usage of illuminating applications, talk based partners and chatbots to automate correspondence and make modified customer experiences at scale. The term "ChatterBot" was at first initiated by Michael Mauldin (producer of the first Verbot, Julia) in 1994 to depict these conversational undertakings. Today, most chatbots are found a good pace of remote partners, for instance, Google Assistant and Amazon Alexa, by methods for illuminating applications, for instance, Facebook Messenger or WeChat, or through individual affiliations' applications and locales. Chatbots can be organized into usage classes, for instance, conversational business (online business by methods for talk), assessment, correspondence, customer help, structure, creator gadgets, preparing, beguilement, finance, sustenance, games, prosperity, HR, publicizing, news, singular, productivity, shopping, social, sports, travel and utilities. Most of the chatbots are furnished with a task individual sort interface with a commitment from a customer and a yield from the chatbot. The chatbot structures the customer's data and yields an answer reliant on what the customer has as of late sent. It could be a welcome, conversation subject, or even an image. Chatbots are normally used in return systems for various sensible purposes including customer help or information acquisition. Most central chatbots work by organizing a customer's commitment with a predefined set of talk. For example, a customer saying "Thank you" will result in the chatbot saying "You're Welcome". Chatbots continue creating in pervasiveness with 80% of associations planning to use one by 2020. Notwithstanding the way that it may feel like the term 'chatbot' has starting late entered the open word reference, they truly have a more drawn out history than you may foresee.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Existing Methodologies.

There are several existing methodologies used for developing chatbots. Here are some of them:

1. **Rule-based chatbots:** These chatbots follow a set of predefined rules to respond to user queries. The rules are written by developers and are specific to the use case. Rule-based chatbots can only answer questions they have been programmed for, and their responses are limited to the set of rules they have been provided.
2. **Retrieval-based chatbots:** These chatbots use predefined responses to match user input. They work by analyzing the user's input and selecting an appropriate response from a pre-defined list of responses. Retrieval-based chatbots use natural language processing (NLP) techniques to understand user input.
3. **Generative chatbots:** These chatbots use machine learning (ML) techniques to generate responses to user input. Unlike retrieval-based chatbots, they do not rely on pre-defined responses but instead generate new responses based on the context of the conversation. Generative chatbots are more advanced than rule-based or retrieval-based chatbots, but they require a significant amount of data and training.
4. **Hybrid chatbots:** These chatbots combine the features of rule-based, retrieval-based, and generative chatbots. They use pre-defined rules for some user queries and generate new responses for others. Hybrid chatbots can switch between different methodologies depending on the context of the conversation.

5. **Self-learning chatbots:** These chatbots use ML algorithms to continuously learn from user interactions and improve their responses over time. They can adapt to new situations and user input, making them more effective and efficient over time.

Each methodology has its advantages and disadvantages, and the choice of methodology depends on the use case and the resources available for development. Rule-based chatbots are easier to develop and require less training data, while generative chatbots are more advanced but require more resources. Hybrid and self-learning chatbots offer a balance between the two.

3.2 Proposed Methodology.

There are several proposed methodologies for chatbots, and here are a few examples:

1. **Natural Language Understanding (NLU):** NLU is the process of analyzing user input and extracting relevant information to understand their intent. This methodology involves the use of machine learning algorithms and semantic analysis techniques to improve the accuracy of the chatbot's responses.

2. **Generative Adversarial Networks (GANs):** GANs are a type of neural network that involves two models - a generator and a discriminator. The generator creates new responses based on the input, while the discriminator evaluates the quality of the response. This methodology can be used to generate more human-like responses.

3. **Reinforcement Learning (RL):** RL is a type of machine learning that involves training the chatbot through trial and error. The chatbot is rewarded for successful interactions and penalized for unsuccessful interactions, which helps it improve over time.

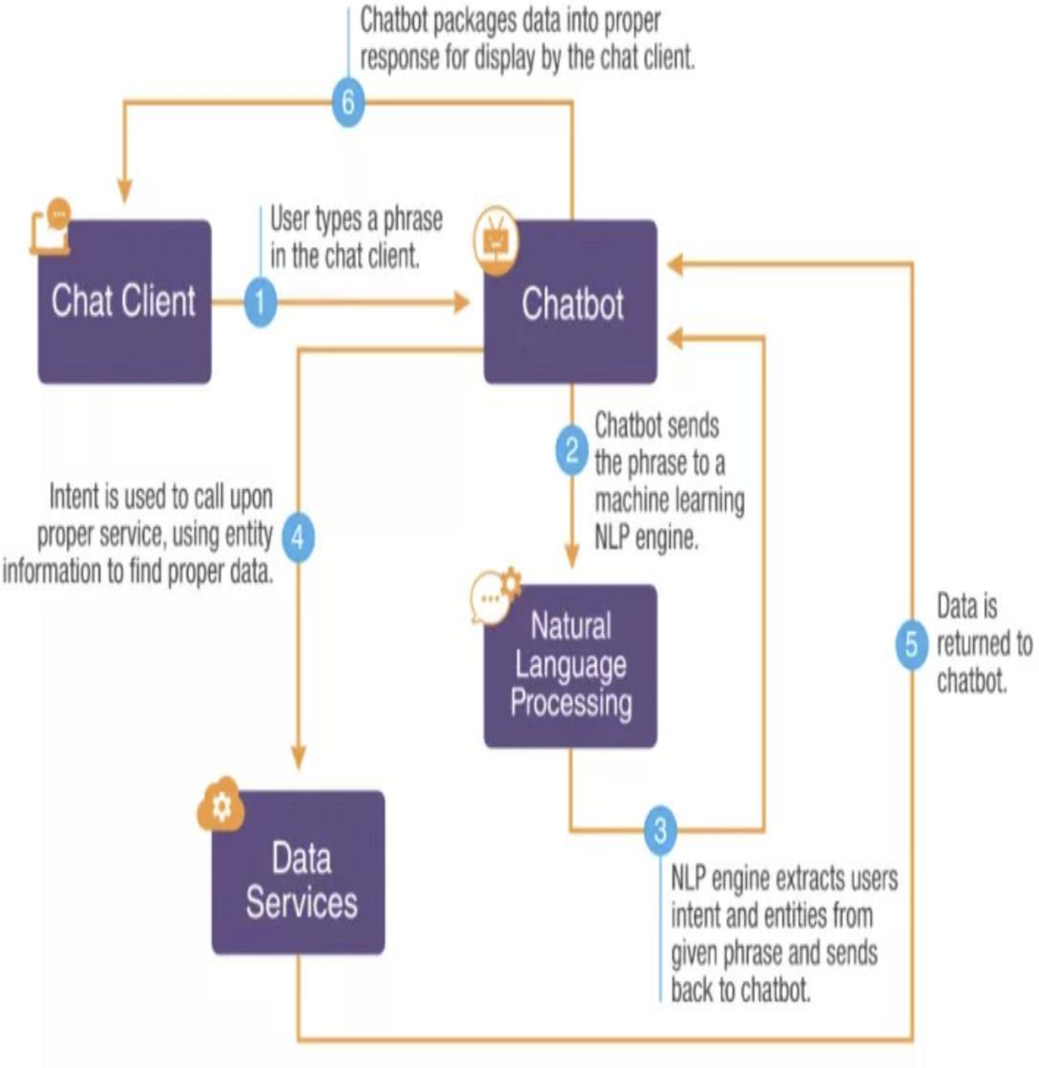
4. **Hybrid Approach:** A hybrid approach involves combining multiple methodologies to create a more effective chatbot. For example, a chatbot may use NLU to understand user input, GANs to generate responses, and RL to improve over time.

5. **Contextual Chatbots:** Contextual chatbots use context and conversation history to generate more personalized responses. This methodology involves using machine learning algorithms to analyze previous interactions and tailor responses based on the user's preferences and history.

6. **Knowledge Graphs:** Knowledge graphs involve using a database of structured information to provide more accurate responses. This methodology involves mapping out relationships between entities and using this information to generate more contextually relevant responses.

These are just a few examples of proposed methodologies for chatbots, and new approaches are constantly being developed as technology advances. Ultimately, the best methodology for a chatbot will depend on the specific use case and the goals of the organization implementing it.

3.1 Architecture of proposed system.


Chatbot packages data into proper response for display by the chat client.
6
Chat Client
User types a phrase in the chat client.
1
Chatbot
2
Chatbot sends the phrase to a machine learning NLP engine.
Intent is used to call upon proper service, using entity information to find proper data.
4
Natural Language Processing
5
Data is returned to chatbot.
Data Services
3
NLP engine extracts users intent and entities from given phrase and sends back to chatbot.

3.3 SYSTEM SPECIFICATION

3.3.1 Software Requirements :

• Operating System	: Windows 10/ 11or updated versions.
• Language	: Python , Angular Js
• Developing Tools	: Visual Studio Code , Notepad
• Technology	: Artificial Intelligence

3.3.2 Hardware Requirements :

• Processor	: Dual Core
• Ram	: 2GB
• Hard Disk	: 128 GB Space

3.3.3 Functional Requirements :

Functional requirements are the capabilities that a chatbot must possess to meet the needs of its users. Some of the essential functional requirements for chatbot development include:

1. **Natural Language Processing (NLP):** Chatbots must have the ability to understand and interpret human language, including its context and intent, to provide relevant responses. NLP technology is used to train chatbots to recognize and respond appropriately to user queries.

2. **Multilingual Support:** With a global audience, chatbots must be able to communicate in multiple languages to cater to users from different parts of the world.

3. **User Authentication:** Chatbots that handle sensitive information such as financial transactions or personal data need to have a secure

authentication process to ensure that only authorized users can access this information.

4. Integration with Other Systems: Chatbots must be able to integrate with other systems such as databases, APIs, and messaging platforms to provide seamless services to users.

5. Personalization: Chatbots must be able to recognize users and personalize their interactions based on previous conversations, preferences, and behavior patterns.

6. Feedback Mechanisms: Chatbots must have a mechanism for collecting user feedback to improve their responses and user experience.

7. Error Handling: Chatbots must have a system for handling errors and providing helpful messages when they encounter a request they cannot understand or fulfill.

8. Continuous Learning: Chatbots must have the ability to learn from user interactions and improve their responses over time. This is typically achieved using machine learning algorithms and techniques.

9. Context Awareness: Chatbots must be able to understand the context of a conversation and provide relevant responses based on the conversation history.

10. Analytics: Chatbots must be able to collect and analyze data on user interactions to provide insights into user behavior, preferences, and needs. This information can be used to improve the chatbot's performance and user experience.

3.3.4 Non Functional Requirements :

In addition to functional requirements, there are several non-functional requirements that need to be considered during chatbot development. These include:

1. Performance: Chatbots need to be highly responsive and provide quick and accurate responses to user queries. Performance requirements may include response time, throughput, and scalability.

2. Reliability: Chatbots need to be reliable and available to users at all times. They should be able to handle large volumes of requests without crashing or experiencing downtime.

3. Security: Chatbots should be designed with robust security measures to prevent unauthorized access, data breaches, and other security threats. This may include encryption of sensitive data, user authentication, and secure communication protocols.

4. Usability: Chatbots should be easy to use and navigate for all users, regardless of their technical expertise. They should have a user-friendly interface and be able to understand natural language inputs.

5. Maintainability: Chatbots should be designed with maintainability in mind, so that they can be easily updated, modified, and maintained over time.

This may involve modular design, use of standard coding practices, and documentation of code.

6. Integration: Chatbots need to be seamlessly integrated with other systems and applications to enable efficient data exchange and workflow management.

7. Accessibility: Chatbots should be designed to be accessible to all users, including those with disabilities or special needs. This may involve providing support for assistive technologies, such as screen readers or voice commands.

By considering these non-functional requirements, chatbot developers can ensure that their chatbots are not only functional, but also reliable, secure, usable, maintainable, and accessible to all users.

CHAPTER 4

MODULES DESCRIPTION

4.1 List of modules :

4.1.1 Pre-processing of input data

4.1.2 Chatterbot

4.1.3 Flask

4.1.4 Flask_cors

4.1.1 Pre-processing of input data :

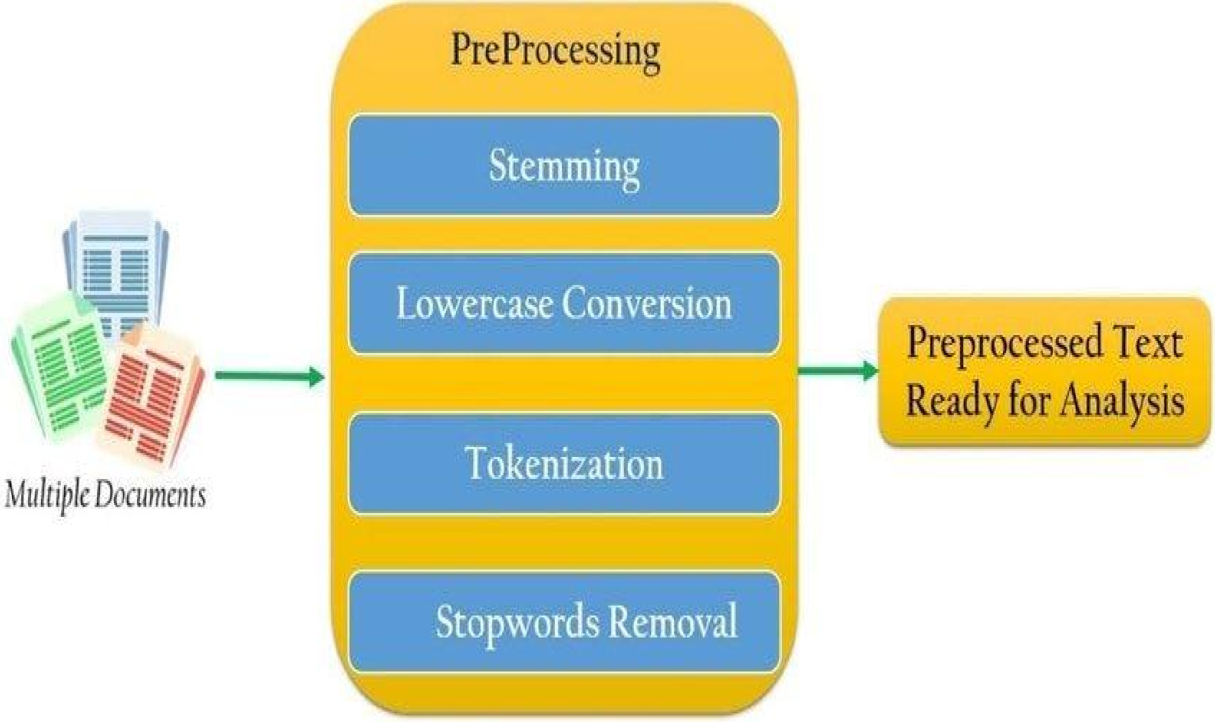
1. Tokenization: The user input text is tokenized into a list of words using the `word_tokenize` function from the `nltk` library. Tokenization is the process of splitting text into individual words or tokens.

2. Stemming: The words in the tokenized input text are stemmed using the `SnowballStemmer` from the `nltk.stem.snowball` module. Stemming is the process of reducing a word to its base or root form.

3. Stop word removal: Stop words are common words such as "the", "a", "an", "in", "at", etc., that do not provide much meaning to a sentence. These words are removed from the tokenized input text using the `stopwords` corpus from the `nltk.corpus` module.

4. Bag-of-words: A bag-of-words is a representation of text that counts the frequency of each word in a document or sentence. The bag-of-words is created using the `Counter` function from the `collections` module.

4.1 Preproceesing of input data



These preprocessing techniques are used to clean and transform the user input text into a format that can be used to find matching responses in the dataset or to generate a response using the chatbot.

4.1.2 Chatterbot :

Chatterbot is a Python library that can be used to develop chatbots. It is an open-source project that provides a framework for creating conversational agents that can interact with users through text or voice interfaces. Here are some key features of Chatterbot:

1. **Language-agnostic:** Chatterbot can be used with any programming language and can support multiple natural languages.

2. **Modular design:** Chatterbot has a modular design that allows developers to easily extend its functionality and add new features.

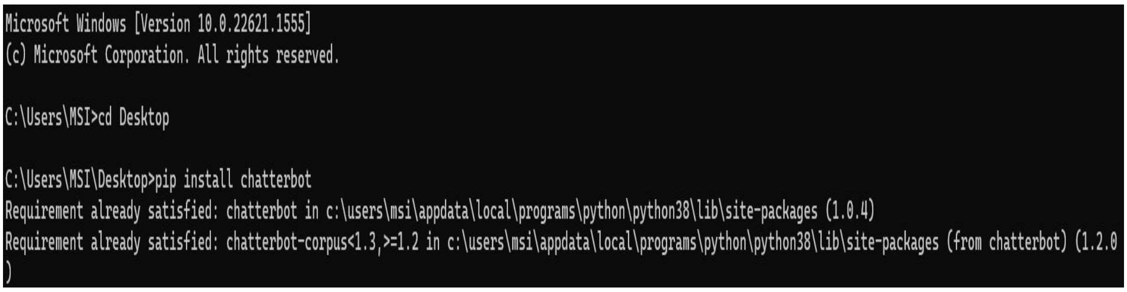
3. **Machine learning:** Chatterbot uses machine learning algorithms to generate responses to user queries. The library includes pre-built machine learning models that can be trained on custom data.

4. **Conversational context:** Chatterbot uses conversational context to generate more natural and relevant responses to user queries. The library keeps track of the conversation history and uses it to generate responses that are consistent with previous interactions.

5. **Open source:** Chatterbot is an open-source project that is actively maintained by a community of developers. This means that developers can access the source code, contribute to the project, and customize the library to fit their specific needs.

To use Chatterbot, developers can install the library using pip and then import it into their Python projects. The library includes several pre-built components, such as the training data and machine learning models, that can be used to quickly build a chatbot. Developers can also customize the library by adding new training data, building new machine learning models, or creating their own conversational logic.

4.2 Installation of chatterbot



```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MSI>cd Desktop

C:\Users\MSI\Desktop>pip install chatterbot
Requirement already satisfied: chatterbot in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (1.0.4)
Requirement already satisfied: chatterbot-corpus<1.3,>=1.2 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from chatterbot) (1.2.0
)
```

4.1.3 Flask :

Flask is a lightweight web framework for Python that is used for building web applications. It is known for its simplicity and flexibility, and it is a popular choice for developers who want to create small to medium-sized web applications quickly and easily.

Here are some key features of Flask:

1. Lightweight: Flask is a lightweight web framework that does not come with a lot of dependencies or unnecessary features. This makes it easy to learn and use, and it also makes the resulting web applications fast and efficient.

2. Flexibility: Flask is highly flexible and allows developers to choose their own tools and libraries to build their applications. This means that

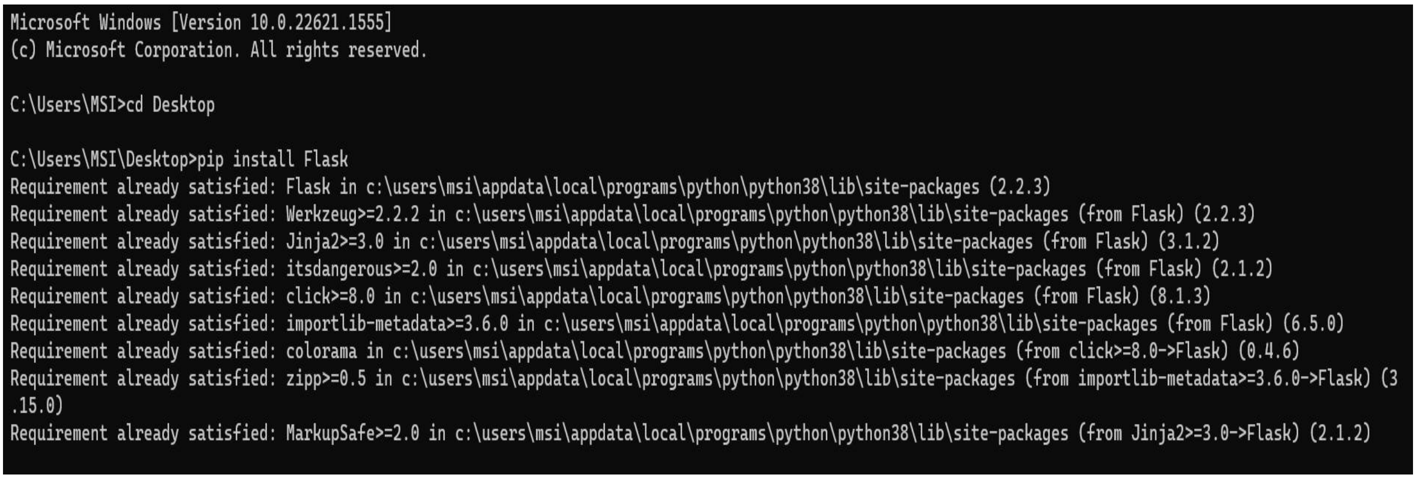
Flask can be used to build a wide range of web applications, from simple static websites to complex web applications with dynamic content.

3. Templating: Flask comes with a built-in templating engine that allows developers to create dynamic HTML pages by inserting data into templates. This makes it easy to build web applications that display data from a database or from user input.

4. Routing: Flask allows developers to define URL routes for their web applications, which determines how users can access different pages or resources. This makes it easy to build RESTful APIs or to handle user input from forms.

5. Extensions: Flask has a large number of extensions available that provide additional functionality, such as database integration, authentication, and caching. These extensions can be easily added to a Flask application to add new features and capabilities.

4.3 Installation of flask



```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MSI>cd Desktop

C:\Users\MSI\Desktop>pip install Flask
Requirement already satisfied: Flask in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (2.2.3)
Requirement already satisfied: Werkzeug>=2.2.2 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Flask) (2.2.3)
Requirement already satisfied: Jinja2>=3.0 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Flask) (2.1.2)
Requirement already satisfied: click>=8.0 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Flask) (8.1.3)
Requirement already satisfied: importlib-metadata>=3.6.0 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Flask) (6.5.0)
Requirement already satisfied: colorama in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from click>=8.0->Flask) (0.4.6)
Requirement already satisfied: zipp>=0.5 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from importlib-metadata>=3.6.0->Flask) (3
.15.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Jinja2>=3.0->Flask) (2.1.2)
```

Flask is a popular choice for building web applications because it is easy to learn and use, and it is highly customizable. It is well-suited for building small to medium-sized web applications, but it can also be used to build larger web applications with more complex functionality.

4.1.4 Flask-cors :

Flask-Cors is a Flask extension that provides Cross-Origin Resource Sharing (CORS) support for Flask applications. CORS is a security feature implemented by web browsers that restricts web pages from making requests to a different domain than the one that served the web page. CORS is important for web security, but it can sometimes restrict legitimate requests between different domains.

Flask-Cors provides a simple way to enable CORS support for a Flask application by allowing requests from specified origins. This means that a Flask application can receive requests from a different domain than the one that served the application.

Here are some key features of Flask-Cors:

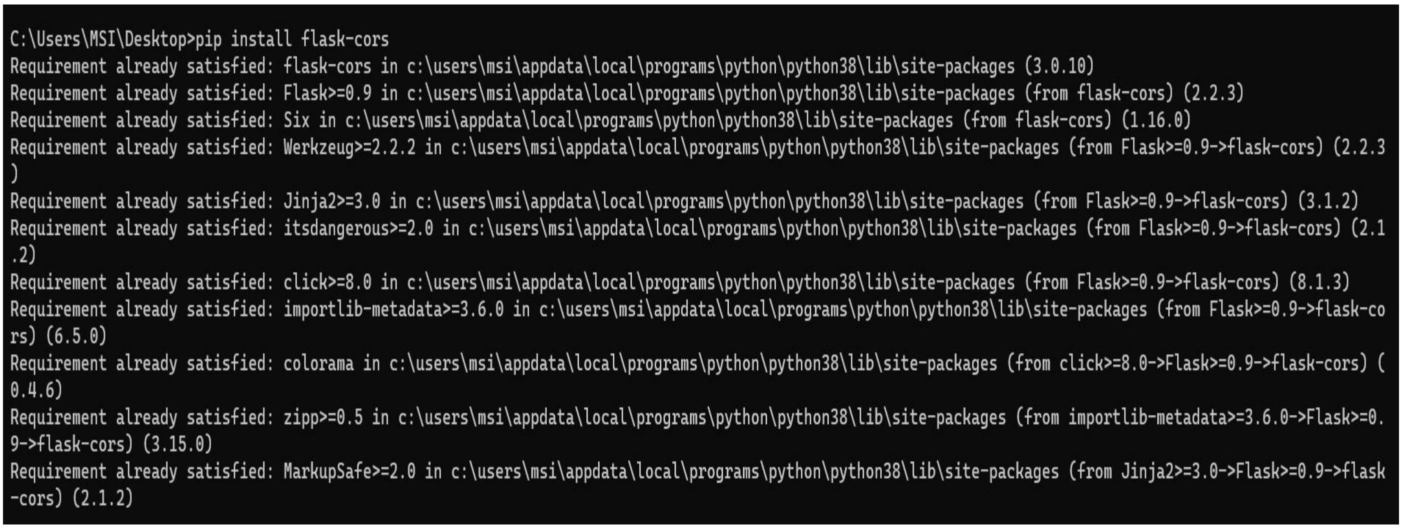
1. **Simple configuration:** Flask-Cors can be easily configured by adding a few lines of code to a Flask application. Developers can specify the allowed origins, headers, methods, and cookies that are allowed to be used with the application.

2. **Multiple origins support:** Flask-Cors allows developers to specify multiple origins that are allowed to make requests to the Flask application. This makes it easy to build web applications that can be accessed from multiple domains.

3. Customizable: Flask-Cors provides a lot of customization options, including specifying the maximum age of cached preflight requests, enabling support for non-simple requests, and specifying the maximum number of headers that can be included in a request.

4. Error handling: Flask-Cors provides error handling for CORS-related errors, which makes it easy to debug issues with CORS requests.

4.4 Installation of flask_cors



```
C:\Users\MSI\Desktop>pip install flask-cors
Requirement already satisfied: flask-cors in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (3.0.10)
Requirement already satisfied: Flask>=0.9 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from flask-cors) (2.2.3)
Requirement already satisfied: Six in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from flask-cors) (1.16.0)
Requirement already satisfied: Werkzeug>=2.2.2 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Flask>=0.9->flask-cors) (2.2.3
)
Requirement already satisfied: Jinja2>=3.0 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Flask>=0.9->flask-cors) (3.1.2)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Flask>=0.9->flask-cors) (2.1
.2)
Requirement already satisfied: click>=8.0 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Flask>=0.9->flask-cors) (8.1.3)
Requirement already satisfied: importlib-metadata>=3.6.0 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Flask>=0.9->flask-co
rs) (6.5.0)
Requirement already satisfied: colorama in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from click>=8.0->Flask>=0.9->flask-cors) (
0.4.6)
Requirement already satisfied: zipp>=0.5 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from importlib-metadata>=3.6.0->Flask>=0.
9->flask-cors) (3.15.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\msi\appdata\local\programs\python\python38\lib\site-packages (from Jinja2>=3.0->Flask>=0.9->flask
-cors) (2.1.2)
```

Overall, Flask-Cors is a useful Flask extension for developers who want to enable CORS support in their Flask applications. It is easy to use, highly customizable, and provides error handling for CORS-related issues.

CHAPTER 5

APPLICATIONS

5.1 SERVER INTERFACE APPLICATION

5.1.1 Flask

Flask is a popular Python web framework used for building web applications. In this code, Flask is used to create a web API endpoint that receives incoming requests from a chatbot interface (such as a web chat interface) and responds with an appropriate response generated by the chatbot.

The Flask app is created using the following lines of code:

```
app = Flask(__name__)
CORS(app, resources={r"/*": {"origins": "http://localhost:4200"}})
@app.route('/chatbot', methods=['POST'])
def chatbot_response():
# Get the user input from the request
inputText = request.json['inputText']
# ... code to generate chatbot response ...
# Return the chatbot response as a JSON object
return jsonify({'bot_response': bot_response})
```

Here, the Flask class is instantiated and assigned to the app variable. The CORS library is also used to enable cross-origin resource sharing, allowing the web API to be accessed from other domains.

The @app.route() decorator is then used to create a route that handles incoming requests to the /chatbot endpoint. This endpoint receives incoming requests from the chatbot interface and returns the chatbot's response to the user.

In this function, the request module is used to retrieve the user's input text from the incoming request, and the generated chatbot response is returned as a JSON object using the jsonify() method from the flask module.

5.1.2 Python

Python is a popular programming language used for developing chatbots due to its simplicity, versatility, and rich ecosystem of libraries and frameworks. Python provides a variety of natural language processing (NLP) libraries such as NLTK, spaCy, and TextBlob that can be used for performing various NLP tasks such as text preprocessing, tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, and more. These libraries provide an easy-to-use interface for working with text data, allowing developers to build chatbots that can understand and respond to user queries in a natural and intuitive way.

In addition to NLP libraries, Python also provides a number of frameworks for building chatbots, such as ChatterBot, Rasa, and BotStar. These frameworks provide a higher-level abstraction for building chatbots and typically include pre-built modules for handling common chatbot tasks such as natural language understanding (NLU), dialogue management, and response generation. They also often provide built-in support for various messaging platforms such as

Facebook Messenger, Slack, and Telegram, making it easy to deploy chatbots to these platforms without having to write custom code.

Overall, Python provides a powerful and flexible platform for building chatbots, allowing developers to leverage a wide range of NLP and machine learning tools and frameworks to create chatbots that can understand and respond to user queries in a natural and intelligent way.

5.2 UI APPLICATION

5.2.1 Angular

Angular is a popular front-end web development framework that allows developers to build dynamic and complex single-page web applications. It is maintained and developed by Google and is one of the most widely used web development frameworks.

Angular is based on TypeScript, a typed superset of JavaScript that provides additional features such as interfaces, classes, and modules. TypeScript makes it easier to develop and maintain large-scale applications by catching common errors and providing better code organization and structure.

Angular provides a powerful set of features and tools that enable developers to create scalable and maintainable web applications, including:

1.Components and directives: Angular uses a component-based architecture, where each component is a self-contained unit of functionality that includes its own HTML template, styles, and logic.

2.Dependency injection: Angular's dependency injection system makes it easy to manage dependencies between components and services, and allows for better testing and reusability of code.

3.RxJS: Angular uses RxJS, a library for reactive programming, to handle asynchronous events and data streams.

4.Forms: Angular provides a powerful set of tools for building and validating forms, including template-driven and reactive forms.

5.Routing: Angular's built-in router provides a powerful and flexible system for handling client-side routing in single-page applications.

Overall, Angular is a versatile and powerful framework that is well-suited for building complex, data-driven web applications. Its rich set of features and tools make it a popular choice for developers and companies looking to build high-quality web applications quickly and efficiently.

5.3 DATABASE

5.3.1 MS EXCEL

Microsoft Excel is used to read the conversation data for the chatbot. Specifically, the `pandas` library in Python is used to read an Excel file located at `C:/Users/MSI/Desktop/Book.xlsx`. The `read_excel` function in `pandas` is used to read the file and create a DataFrame object. The `dtype` parameter is used to specify the data type of the "Input" column in the Excel file as string.

Next, the DataFrame is converted to a list of tuples containing the conversation data, which is then used to train the chatbot. The conversation data in Excel contains two columns: "Input" and "Response". The "Input" column contains user inputs, while the "Response" column contains the corresponding responses from the chatbot.

Overall, Microsoft Excel is used as a data source for the chatbot, and Python libraries such as `pandas` are used to read and manipulate the data.

CHAPTER 6

EXPERIMENTAL RESULT AND ANALYSIS

The testing of chatbot's performance in responding to user inputs, evaluating its accuracy and efficiency, and analyzing the data collected from those interactions. This could involve collecting user feedback, monitoring chatbot performance metrics, and using statistical methods to analyze the data and draw conclusions about the chatbot's effectiveness and following functions can provide the includes like,

- Providing information about the college's academic programs, admission requirements, and application process.
- Answering frequently asked questions (FAQs) about the college, such as the location, facilities, tuition fees, and financial aid options.
- Providing support services for students, such as counseling, academic advising, and career guidance.
- Helping students connect with faculty members, alumni, and peer mentors.
- Conducting surveys and collecting feedback from students to improve college services and programs.
- Providing updates and announcements about college news, events, and initiatives.
- Personalizing the chatbot interactions based on the student's profile, preferences, and history.
- Integrating with other college systems, such as the student information system (SIS), learning management system (LMS), and library system.

CHAPTER 7

CODE

7.1 Front-end:

chatbot.component.html:

```
 <div class="chatbot-container">
<div class="chatbot-header">
 <img src = "../../assets/image/logo.png">
 <h3> Coimbatore Institute Engineering
      and Technology</h3>
</div>
<div class="chatbot-messages">
 <div class="message" *ngFor="let message of messages">
   <div class="message-text" [ngClass]="{'user': message.sender === 'user', 'chatbot': message.sender
=== 'chatbot'}">
     {{ message.text }}
   </div>
 </div>
</div>
<div class="chatbot-input">
 <input type="text" [(ngModel)]="inputText"  placeholder="Type your message here">
 <button (click)="sendMessage()">Send</button>
</div>
</div>
```

chatbot.component.css:

```
.chatbot-container {
   width: 400px;
   height: 500px;
   border-radius: 10px;
   box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
   border: 5px solid #3e423f;
   display: flex;
   flex-direction: column;
   overflow: hidden;
   position: fixed;
   top: 50%;
   left: 50%;
   transform: translate(-50%, -50%);
 }
 .chatbot-header img {
   width: 20%;
   float: left;
   margin-right: 10px;
   padding-left: 10px;
 }
 .chatbot-header {
   background-color:  #11309f;
```

```
    color: white;

    font-size: 24px;

   padding-top:  12px;

    text-align: center;

  }

  .chatbot-messages {

    flex-grow: 1;

    overflow-y: auto;

    padding: 20px;

     }

  .message {

    margin-bottom: 10px;

    }

  .message-text {

    border-radius: 10px;

    padding: 10px;

    max-width: 70%;

    }

   .user {

background-color: #f1f1f1;

    margin-left: auto;

  }

  .chatbot {
    background-color:  #11309f;
    color: white;
    margin-right: auto;
  }
.chatbot-input {
    display: flex;
    padding: 20px;
  }
  input[type="text"] {
    flex-grow: 1;
    border-radius: 5px;
    border: none;
    padding: 10px;
    font-size: 16px;
    background-color: #eff0e9e6;
  }
button {
    background-color: #11309f;
    color: white;
    border-radius: 5px;
    border: none;
    padding: 10px 20px;
    margin-left: 10px;
    cursor: pointer;

    font-size: 16px;

  }

 button:hover {

    background-color:  #11309f;

  }
```

chatbot.component.ts:

```
import { Component } from '@angular/core';

import { HttpClient } from '@angular/common/http';

@Component({

  selector: 'app-chatbot',

  templateUrl: './chatbot.component.html',

  styleUrls: ['./chatbot.component.css']

})

export class ChatbotComponent {

  messages: { text: string, sender: string }[] = [];

  inputText = '';

constructor(private http: HttpClient) { }

sendMessage() {

    if (this.inputText.trim()) {

      this.messages.push({ text: this.inputText, sender: 'user' });

      this.http.post<any>('http://localhost:5000/chatbot', { inputText: this.inputText })

      .subscribe(response => {
```

```
      this.messages.push({ text: response.bot_response, sender: 'chatbot' });
    }, error => {
      console.error('Error sending message:', error);
    });
    console.log('Message sent:', this.inputText);
    this.inputText = '';
   }
  }
 }
```

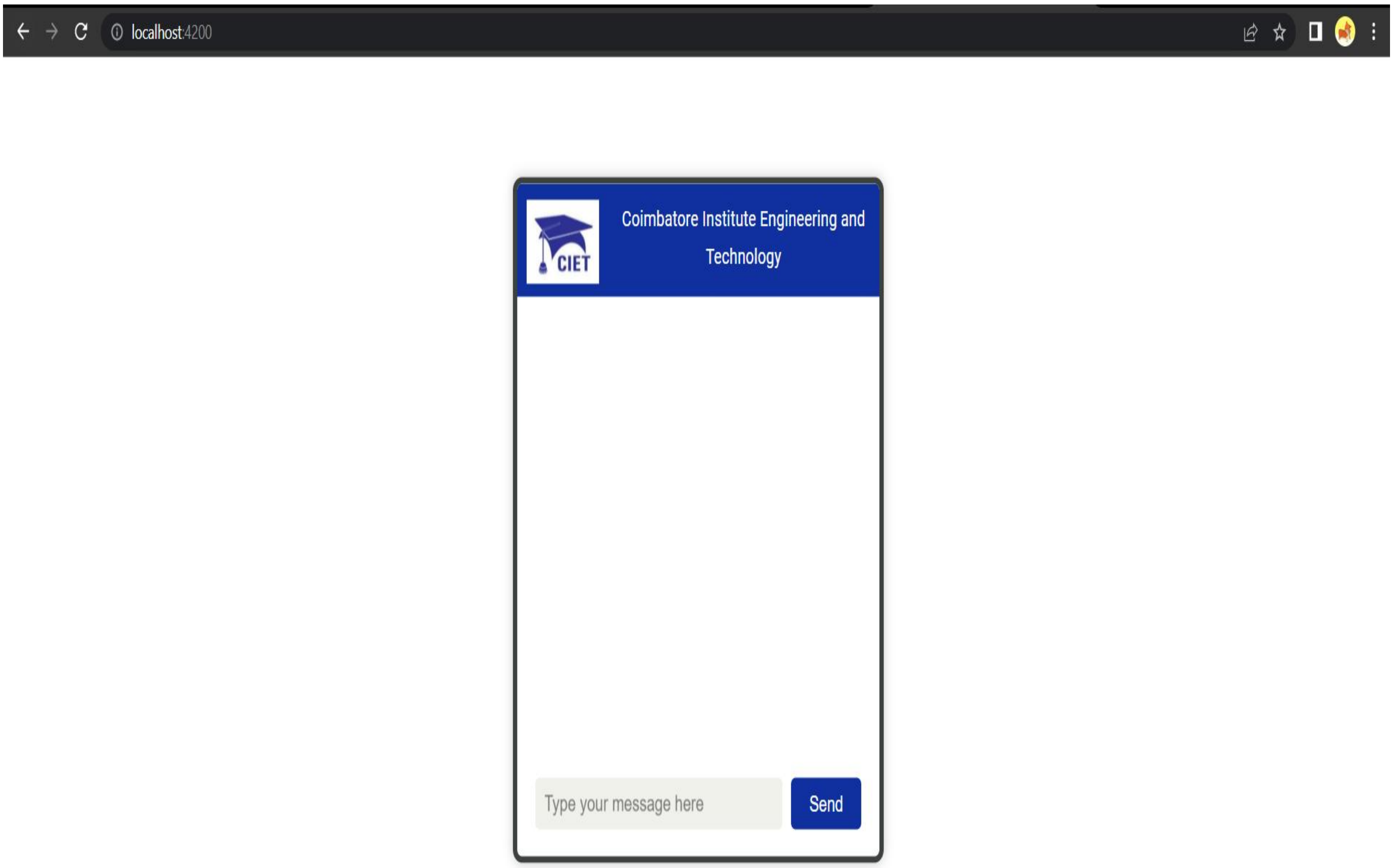
7.2 Back-end:

```
import pandas as pd
import nltk
from nltk.stem.snowball import SnowballStemmer
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from collections import Counter
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer
from flask import Flask, request, jsonify
from flask_cors import CORS
import time
```

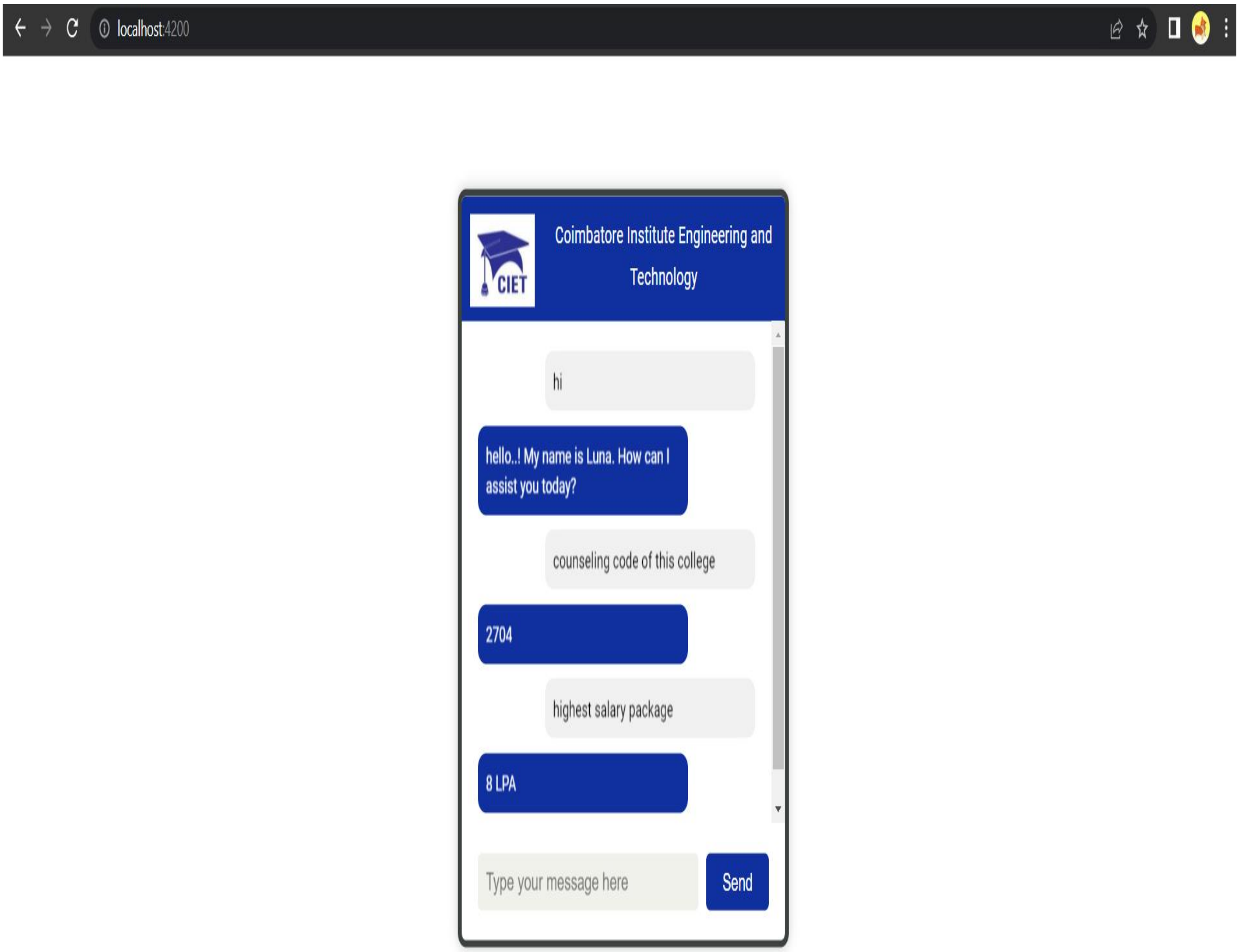
```
# Read the Excel sheet data and convert the "Input" column to string type
df = pd.read_excel('C:/Users/MSI/Desktop/Book.xlsx', dtype={'Input': str})

# Convert the data into a list of tuples containing (input, response) pairs
conversations = df[['Input', 'Response']].apply(tuple, axis=1).tolist()
# Create a chatbot instance
chatbot = ChatBot('MyChatBot')
# Train the chatbot using the conversations from the Excel sheet
trainer = ChatterBotCorpusTrainer(chatbot)
trainer.train('chatterbot.corpus.english')
# Define stemmer and stop words
stemmer = SnowballStemmer("english")
stop_words = set(stopwords.words("english"))
# Create a Flask app
app = Flask(__name__)
CORS(app, resources={r"/*": {"origins": "http://localhost:4200"}})
@app.route('/chatbot', methods=['POST'])
def chatbot_response():
    # Get the user input from the request
    inputText = request.json['inputText']
    #print('Input text:', inputText)
    #Preprocess user input
    tokens = word_tokenize(inputText.lower())
    stemmed_tokens = [stemmer.stem(token) for token in tokens if token not in stop_words]
    bow = Counter(stemmed_tokens)
    # Find the most common words in the bag of words
    common_words = [word for word, count in bow.most_common(5)]
    # Match user input with "Input" column in the Excel sheet
    max_percentage_match = 0
    matched_input = None
    for row in conversations:
        input_text = str(row[0]).lower()
        input_tokens = word_tokenize(input_text)
        input_stemmed_tokens = [stemmer.stem(token) for token in input_tokens if token not in stop_words]
        input_bow = Counter(input_stemmed_tokens)
        common_tokens = set(input_stemmed_tokens) & set(stemmed_tokens)
        if len(common_tokens) > 0:
            percentage_match = sum([input_bow[token] for token in common_tokens]) / sum(input_bow.values())
            if percentage_match > max_percentage_match:
                max_percentage_match = percentage_match
                matched_input = input_text
                bot_response = row[1]
    if matched_input is None:
        bot_response = "Sorry, I can't understand. Could you please provide more information about that?"

    # Return the chatbot response as a JSON object

    return jsonify({'bot_response': bot_response}

if __name__ == '__main__':

    app.run(debug=True)
```

7.3 UI SCREEN SHOTS:

7.1 Displaying UI of chatbot



7.3 Execution of chatbot



Coimbatore Institute Engineering and Technology

hi

hello..! My name is Luna. How can I assist you today?

counseling code of this college

2704

highest salary package

8 LPA

CHAPTER 8

CONCLUSION AND FUTURE WORK

In conclusion, developing a chatbot can bring numerous benefits to a business or organization, such as increased efficiency, cost savings, and improved customer experience. However, it's important to carefully consider the purpose and scope of the chatbot, as well as the target audience and their needs, in order to create a chatbot that truly adds value. The design and development process should prioritize user-centered design principles, and the chatbot should be continuously tested and refined to ensure that it meets the intended goals and delivers a positive experience for users. Additionally, it's important to have a plan in place for ongoing maintenance and updates to ensure the chatbot remains relevant and effective over time. Overall, developing a chatbot can be a valuable investment for businesses and organizations looking to improve customer support, increase efficiency, and reduce costs. However, it's important to approach the process with careful planning and execution to ensure the chatbot meets its intended goals and delivers a positive user experience.

Future work :

The promising future work of this paper is, That we will be focusing on adding the chatbot to the website and regularly analyze the chatbot for its upgradation and update the database with the queries that chatbot can’t provide responses.

CHAPTER 9

REFERENCES

[1] A Review on College Enquiry Chatbot ,Department of Computer G.H. Raisoni College of Engineering and Management, Pune, India

[2] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, Chabot for university-related FAQs,2017 International Conference on Advances in Computing, Communications, and Informatics(ICACCI), Udupi, 2017

[3] Kanchan Chopde, Sanket Mondhe, Nikita Sahu,Shuklesh Deshbhratar, Ms. Priyanka B. Dongre PCE College Enquiry Bot, Priyadarshini College of Engineering, Nagpur, India, International Journal of Innovations in Engineering and Science

[4] Payal Jain, College Enquiry ChatBot Using Iterative Model, International Journal of Scientific Engineering and Research (IJSER)

[5] Sagar Pawar, Omkar Rane, Ojas Wankhade Pradnya Mehta, A Web Based College Enquiry Chatbot with Results, InternationalJournal of Innovative Research in Science, Engineering and Technology

[6] Ms.Ch.Lavanya Susanna, R.Pratyusha, P.Swathi, P.Rishi Krishna, V.SaiPradee, College Enquiry Chatbot, International Research Journal of Engineering and Technology (IRJET)

[7] Intelligent smart chatbot system for college, International Research Journal of Modernization in Engineering Technology and Science

[8] College Enquiry Chatbot System using Artificial Intelligence, International Journal of Scientific Research in Computer Science, Engineering and Information Technology

[9] A Review on College Enquiry Chatbot,G.H. Raisoni College of Engineering and Management, Pune, India.

[10] Research Paper on Chatbot for Student Admission Enquiry,Journal of Advancement in Software Engineering and Testing.